Jr. (U.S. Patent 6,339,838) and Grover (U.S. Patent 6,421,349). It is the Examiner's position that Gall et al teaches a method of performing a system reverse engineering process on an application system which includes the steps of "from the information gathered and recorded by the examination of the network structure forming the application system, formatting the information gathered into a form in which it represents the application system in a usable form". The Examiner acknowledges that Gall et al does not expressly teach the claimed limitation of gathering the entire application system that requires reverse engineering and identifying each development environment associated with the application system.

The Examiner relies on the teachings of Iyengar et al to teach gathering the entire application system that requires reverse engineering and identifying each development environment associated with the application system because that allows storing the output data, application components and the relationship between entities and objects to be stored in the repository, in this way the system integrates the tools used in different parts of the development process by passing necessary information from one tool to another through the repository. The Examiner then states that it would have been obvious to one of ordinary skill in the art at the time of Applicants'

invention to modify the method of Gall et al with the method of Iyengar et al as such a modification would allow storing the output data, application components and the relationship between entities and objects to be stored in the repository, in this way the system would integrate the tools used in different parts of the development process by passing necessary information from one tool to another through the repository.

The Examiner also states in the rejection of the claims that Gall et al teaches a suitably programmed processing system, which is provided with a data base of object types in respect of which the nature, characteristics and properties are known. The Examiner once again acknowledges the deficiencies of Gall et al in that Gall et al does not expressly teach the claimed limitation of a suitable programmed processing system, which is provided with a data base of object types in respect of which the nature, characteristics and properties are known and which fall in groups that includes process or activity control elements, data management elements and interface elements.

The Examiner then turns to Weinman, Jr. as teaching a suitably programmed processing system, which is provided with a data base of object types in respect of which the nature, characteristics and properties are known and which fall in groups that include process or activity control elements, data

management elements and interface elements because that allows interacting with an existing process (EP) to develop information that can be used to enhance EP process model; the modeling tool allows for modifying the model data; the code generation tool allows for the provision and delivery of an enhanced code to the EP process allowing an improved EP process.

The Examiner then states that it would have been obvious to modify the method of Gall et al with the method of Weinman, Jr. as it would allow interacting with an existing process (EP) to develop information that could be used to enhance EP process model; the modeling tool would allow for modifying the model data; the code generation tool would allow for the provision and delivery of an enhanced code to the EP process allowing an improved EP process.

The Examiner also acknowledges that Gall et al. does not expressly teach identifying the object types, as determined by each development environment identified as being associated with the application system, that can serve as starting points from where an examination of the application system can be initiated. The Examiner relies on Grover that teaches identifying the object types, as being associated with the application system, that can serve as starting points from where an examination of the application system can be initiated, because that allows

searching for and identifying a set of intermediate nodes, that together with the originating node may form a path or chain. The Examiner concludes that it would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of Gall et al. with the method of Grover because that would allow searching for and identifying a set of intermediate nodes, that together with the originating node could form a path or chain.

The Examiner additionally states that Gall et al. teaches identifying points, in the form of object instances of object types but again acknowledges that Gall et al. does not expressly teach identifying entry points, in the form of object instances of object types identified to serve as starting points from occurrence of failure of one of the links or nodes of the network. The Examiner once again concludes that it would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of Gall et al with the method of Grover that included examining from selected entry points the network structure forming the application system by tracking chains of nodes and links, each chain being tracked until the instance of a node that does not have a link or the return of the chain to a previously examined node because that would provide a method of establishing a connected route through

the network and would allow restoring networks upon occurrence of failure of one of the links or nodes of the network.

Additionally, the Examiner notes that Gall et al does not expressly teach reverse tracking the chain to a node from which another chain extends and selectively tracking said other chain. The Examiner relies on Grover to teach reverse tracking the chain to a node from which another chain extends and selectively tracking said other chain, because that provides a method of establishing a restoration path for a failed span and allows restoring networks upon occurrence of failure of one of the links or nodes of the network. The Examiner once again concludes that it would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of Gall et al with the method of Grover because that would provide a method of establishing a restoration path for a failed span and would allow restoring networks upon occurrence of failure of one of the links or nodes of the network.

It is also noted by the Examiner that Gall et al does not expressly teach continuing the process until all the chains within the network structure have been tracked completely, the tracking of the chains including an examination of each node and each link in the network structure, to the extent that the nature, characteristics and properties of each node and each

link can be associated with object types through analysis and understanding thereof, and gathering and recording all the information of each node and each link. Grover teaches continuing the process until all the chains within the network structure have been tracked completely, the tracking of the chains including an examination of each node and each link in the network structure, to the extent that the nature, characteristics and properties of each node and each link can be associated with object types through analysis and understanding thereof, and gathering and recording all the information of each node and each link, because that provides a method of establishing a connected route through the network and allows restoring networks upon occurrence of failure of one of the links or nodes of the network. The Examiner again concludes that it would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of Gall et al with the method of Grover because that would provide a method of establishing a connected route through the network and would allow restoring networks upon occurrence of failure of one of the links or nodes of the network.

As per claims 3, 4 and 9, it is the Examiner's position that the combination of Gall et al., Iyengar et al., Weinman, Jr., and Grover renders these claims obvious.

Claims 2, 7, 8 and 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gall et al. ("Balancing in reverse engineering in object oriented systems engineering to improve reusability and maintainability", IEEE, 1994) in view of Iyengar et al. (U.S. Patent 6,018,627), Weinman, Jr. (U.S. Patent 6,339,838) and Grover (U.S. Patent 6,421,349), and further in view of Zgarba et al. (U.S. Patent application 2002/0170048).

The Examiner acknowledges that Gall et al does not expressly teach identifying each development environment associated with the application system, identifying aspects of each development environment selected from the mechanisms of storage of data, the interface of the above, component libraries and code management systems. Iyengar et al. teaches identifying each development environment associated with the application system, identifying aspects of each development environment selected from the mechanisms of storage of data, the interface of the above, component libraries and code management systems, because that allows storing the output data, application components and the relationship between entities and objects to be stored in the repository; in this way the system integrates the tools used in different parts of the development process by passing necessary information from one tool to another through the repository; and code management systems allow the resulting

component code to be versioned using off-the-shelf code management tools. The Examiner concludes that it would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of Gall et al with the method of Iyengar et al because that would allow storing the output data, application components and the relationship between entities and objects to be stored in the repository; in this way the system would integrate the tools used in different parts of the development process by passing necessary information from one tool to another through the repository; and code management systems would allow the resulting component code to be versioned using off-the-shelf code management tools.

The Examiner also acknowledges that Gall et al does not expressly teach identifying each development environment associated with the application system, identifying aspects of each development environment selected from a group including programming language and syntax used. Zgarba et al is relied upon as teaching identifying each development environment associated with the application system, identifying aspects of each development environment selected from a group including programming language and syntax used, because that allows source code written in a particular programming language to be reverse engineered into a new software model and then forward engineered

from software model into a new set of source code; and the software model to be kept up to date with the changing source code allowing generation of easily understood diagrammatic representations of the software. The Examiner concludes that it would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of Gall et al with the method of Zgarba et al because that would allow source code written in a particular programming language to be reverse engineered into a new software model and then forward engineered from software model into a new set of source code; and the software model to be kept up to date with the changing source code allowing generation of easily understood diagrammatic representations of the software.

Claims 7, 8 and 10 are rejected based on the same reasoning as Claims 3, 4 and 9., Claims 7, 8 and 10 are method claims reciting the same limitations as Claims 3, 4 and 9, as taught throughout by Gall et al., Iyengar et al., Weinman, Jr., Grover and Zgarba et al.

Claims 5 and 12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gall et al. ("Balancing in reverse engineering in object oriented systems engineering to improve reusability and maintainability", IEEE, 1994) in view of Iyengar et al. (U.S. Patent 6,018,627), Weinman, Jr. (U.S. Patent

6,339,838) and Grover (U.S. Patent 6,421,349), and further in view of Litt et al. (U.S. Patent 6,635,469) and Digalakis et al. (U.S. Patent 6,256,607).

The Examiner acknowledges that Gall et al does not expressly teach formatting which includes breaking structures into candidate components by using affinity analysis. Litt et al is cited as teaching formatting which includes breaking structures into candidate components by using affinity analysis because that provides a separation technique using physical size or by affinity extraction. The Examiner concludes that it would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of Gall et al with the method of Litt et al because that would provide a separation technique using physical size or by affinity extraction.

The Examiner also acknowledges that Gall et al does not expressly teach formatting that includes breaking structures into candidate components by using mathematical clustering techniques. Digalakis et al teaches formatting that includes breaking structures into candidate components by using mathematical clustering techniques, because that allows efficient numerical encoding of data for use in automatic recognition systems. The Examiner concludes that it would have

been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of Gall et ale with the method of Digalakis et al because that would allow efficient numerical encoding of data for use in automatic recognition systems.

Claim 12 is rejected based on the same reasoning as Claim 5. Claim 12 is a method claim reciting the same limitations as Claim 5, as taught throughout by Gall et al., Iyengar et al., Weinman, Jr., Grover, Litt et al. and Digalakis et al.

Claims 11 and 13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gall et al. ("Balancing in reverse engineering in object oriented systems engineering to improve reusability and maintainability", IEEE, 1994) in view of Iyengar et al. (U.S. Patent 6,018,627), Weinman, Jr. (U.S. Patent 6,339,838), Grover (U.S. Patent 6,421,349) and Zgarba et al. (U.S. Patent application 2002/0170048, and further in view of Litt et al. (U.S. Patent 6,635,469) and Digalakis et al. (U.S. Patent 6,256,607).

As per Claims 11 and 13, the Examiner rejects these claims based on the same reasoning as Claim 5.

The Applicant disagrees with the Examiner's position for the following reasons:

The Gall et al reference was discussed in detail during prosecution of the international application at which it was determined that the claims of the present invention met the requirements of novelty, inventive step and industrial applicability.

It is acknowledged that Gall et al does indeed refer to system reverse engineering with the most relevant section of this document that relates to system reverse engineering as such being the section 2.1 entitled "The design recovery step" (page 36) with subsections 2.1.1 and 2.1.2 being effectively excluded as being relevant, since these sections are concerned with specific software engineering document creation.

The need for a system reverse engineering process as herein envisaged is explained on page 1, lines 3-17 of the compete specification as filed in support of the application under examination. In summary, a system reverse engineering process can avoid the need for requiring the replacement of a complete software system that can no longer be easily managed or that is no longer understood due to software system accretion, which can be extremely expensive and technically very complicated. A reverse engineering process particularly permits re-engineering of the software system into a workable and useful format in a relatively cost-effective manner.

The known approach towards system reverse engineering and which identifies the state of the art in respect of system reverse engineering is set out clearly in the document that is referred to in the first paragraph of the section 2.1 of the Gall et al reference. This document is identified as P. Benedusi, A. Cimitile and U. de Carlini. Reverse Engineering Processes, Design Document Production and Structure Charts. The Journal of Systems and Software, 19(3):225-245, November 1992.

As quoted from Section 2 on page 226, entitled "2. The Goals/Models/Tools Paradigm" the first two paragraphs reading as follows (italics inserted):

"At the current state of the art an RE (*reverse engineering)* process must be defined and designed for each use, taking into account the needs it must meet and the environment in which it must run. At a methodological level, the RE process must be guided by the type of documents to be produced and the development and maintenance processes to which the .documents belong. We cannot talk about an RE process without referring it to a well-defined FE *(Forward Engineering)* process in which the RE process products must later be used. Even if we refer only to the production of design-level information and documents, as we will below, our and others' experiences show that the RE methodologies and techniques proposed in the literature each

have many different specification in the real world of software development and maintenance. This suggests that we should define and try out high-level organization paradigms for RE processes in which an organization can learn from design and development of RE tasks and then tailor and apply that paradigm to several needs [28].

Given these needs, we believe that the definition and production of an RE process must be based on the goals/models/tools paradigm."

The important points from the above paragraphs, which clearly reflect the state of the art, are:

1) a reverse engineering process must be defined and designed for each use, taking into account the needs it must meet and the environment in which it must run;

2) the process must be guided by the type of documents to be produced;

3) cannot talk about reverse engineering process without referring to well defined forward engineering process;

4) the definition and production of a reverse engineering process must be based on the goal/model/tools paradigm.

It is apparent therefore, that known system reverse engineering processes are goal orientated and to accomplish those only relevant parts/segments of systems to be reversed

engineered are extracted and examined in order to reverse engineer these parts/segments into a usable form, i.e. to accomplish the goal.

The system reverse engineering process which forms the invention under examination and as defined in claim 1 is fundamentally different to the known processes insofar as it is not goal orientated and is not designed for a particular use which takes into account the needs it must meet and the environment in which it must run. As such, the process of the invention does not include the steps of extracting only parts/segments of the system to be reverse engineered in order to accommodate a particular goal.

The system reverse engineering process of the invention in fact gathers the entire system that requires reverse engineering and then, by following a series of predetermined method steps, examines the entire network structure forming the system to be reverse engineered by tracking chains of nodes and links, each node and link in the entire system being examined to the extent that it can finally be associated with a known object type. Only when the entire system has been so examined and all information gathered in respect of all nodes and links has been recorded, is the information gathered formatted into a form that represents the original application system in a usable form,

still without any particular goal having been considered. The usable form of the application system, which will now be properly understood, can hence be used to accommodate specified requirements or be adapted or modified to permit particularly applications or goals.

The major benefits of the system reverse engineering process of the invention therefor include that a separate reverse engineering process need not be defined or designed for each use while taking into account needs and goals, it is always ensured that an entire system is reverse engineered and parts or segments of an original system is not ignored or deleted, and the final reverse engineered system will not only be useful for accommodating a predetermined goal, but could be associated with and rendered usable in conjunction with a number of different goals.

Clearly, the state of the art reverse engineering processes that are known and the process of the invention are fundamentally different, with the method of the invention being associated with a number of distinct benefits and advantages.

In the rejection, the Examiner acknowledges the numerous deficiencies of the Gall et al reference with respect to the presently claimed invention. In fact, the Examiner relies on no

less than **SIX additional references** to render an "obviousness" decision regarding the present invention.

The reference to Iyengar et al was cited to teach gathering the entire application system that requires reverse engineering and identifying each development environment associated with the application system because that allows storing the output data, application components and the relationship between entities and objects to be stored in the repository, in this way the system integrates the tools used in different parts of the development process by passing necessary information from one tool to another through the repository.

The Applicant disagrees with the use of Iyengar et al as this invention deals with individual development environments in isolation. The automation of the process is inhibited by this non-comprehensive individualistic approach. Iyengar et al's primary approach of dealing with legacy applications is via undocumented 3rd party tools not covered by the reference patent-see col. 8, paragraph 1. The reference patent as a whole is concerned with Application Building in an Object Oriented Development Environment. In direct contrast, the claimed invention is centered around heterogeneous, comprehensive, automated reverse engineering entailing pure non-domain specific nodes and links. The use of a single process/method that is not

domain and goal specific yet resolves all, including redundant nodes and links would be non-obvious to one having ordinary skill in the art. For the reasons set forth above, it is Applicant's position that Iyengar et al, when used in combination with Gall et al, fail to render any portion of the claims obvious.

With respect to the teachings of Weinman, Jr., the Examiner relies on this reference as teaching a suitably programmed processing system, which is provided with a data base of object types in respect of which the nature, characteristics and properties are known and which fall in groups that include process or activity control elements, data management elements and interface elements because that allows interacting with an existing process (EP) to develop information that can be used to enhance EP process model; the modeling tool allows for modifying the model data; the code generation tool allows for the provision and delivery of an enhanced code to the EP process allowing an improved EP process.

The Applicant disagrees with the use of this reference as it is from the field of process control of both industrial and business, and deals with physical workflow models in particular and is concerned with reverse engineering only insofar as it can populate process modeling diagramming conventions-see col. 1,

lines 4-6 and col. 2. The use of the term "reverse engineering" is common but non-analogous in respect of both the context and interpretation in that Weinman, Jr. is preoccupied with the generation of graphical process modeling language and programming code-col. 14, paragraph 4. Weinman, Jr. essentially claims an arrangement comprising: a repository containing a process model and a set of 3rd party tools that interact with the repository. The generalized narrative of Weinman, Jr. serves only to cloud its claims, which are centered around industrial and business process control. Nowhere is a generic method of reverse engineering invented.

With respect to the teachings of Grover, this reference deals entirely with the physical world networks of physical objects within a telecommunications network context and thus it is the Applicant's position that the reliance on this references is based purely on hindsight reasoning. The references relied upon by the Examiner are taken out of context and are misread. The Examiner describes selecting specific origin nodes that provide the discovery of "closed" paths, indicating a return path to the originating node. This is non-analogous and dissimilar to the claimed invention's use and context of entry points, nodes and links-see col. 3, lines 17-20, col. 45, paragraphs 3-5 and col. 46. The references description of

broadcasting from an originating node utilizing an originating statelet is read out of context. This in non-analogous and dissimilar to the claimed invention's use and context of navigating nodes and links, the singular or parallel navigations begun by the claimed invention is the systematic traversal of the inherent network of nodes and links, not a general broadcast method. The Grover node traversal mechanism in non-analogous further to the claimed invention in that in the process of the discovery of a closed path network one and only one link between two nodes is mapped and mapped only once, whereas in the claimed invention, the systematic navigation discovers all links, forward and reverse (including apparently redundant nodes and links), between all nodes and in so doing may discover a multiplicity of links between any two nodes either inherent or implied-see col. 3, lines 33-34, col. 3, paragraphs 5-6.

For the reasons set forth above, it is the Applicant's position that the addition of Grover to the rejection, in combination with Gall et al, fail to render any portion of the claims obvious.

With respect to the teachings of Zgarba et al, it is Applicant's position that this reference is non-analogous as it is whole objective driven in respect of round trip engineering and within a specific domain-see page 1, paragraphs 1 and 8.

Zgarba et al also is a non-comprehensive reversal of application based on the restrictions of the goal oriented software model-page 3, paragraph 38. The focus of the reference invention is the process activity side of an application, but does not encompass applications in respect of a heterogeneous nature with respect to their development environments. Therefore non-analogous to the claimed invention which reverse engineer the entire application including, but not limited to process logic, data structure and even data itself-see page 2, paragraph 28. The data model elements of applications are excluded as this would not form part of a roundtrip engineering code version control as per the stated referenced invention as a whole. Furthermore, the intended physical output of the referenced patent is the same augmented source code as the input source code, both in development environment and target application terms. The referenced patent involves a single development environment at a time to augment the forward code development process within the particular development environment selected at that time only-page 3, paragraph 35.

For the reasons set forth above, it is the Applicant's position that the addition of Zgarba et al to the rejection, in combination with Gall et al, fail to render any portion of the claims obvious.

With respect to the teachings of Litt et al, it is the Applicant's position that this patent is outside of the field of the claimed invention. Biomolecular complexes of human tissue are completely irrelevant to the invention. References within the claimed invention to affinity analysis serve as examples of analysis types (in that there are many other entirely different kinds of analyses possible) to illustrate the utility of the now discovered application system within a usable form.

For the reasons set forth above, it is the Applicant's position that the addition of Litt et al to the rejection, in combination with Gall et al, fail to render any portion of the claims obvious.

With respect to the teachings of Digalakis et al, it is Applicant's position that this references is also outside the field of the claimed invention. As a whole, it is a novel approach for encoding signals and thus the whole invention is non-analogous. References within the claimed invention to clustering serve as examples of analysis types (in that there are many other entirely different kinds of analyses possible) to illustrate the utility of the now discovered application system within a usable form.

For the reasons set forth above, it is the Applicant's position that the addition of Digalakis et al to the rejection,

in combination with Gall et al, fail to render any portion of the claims obvious.

For the reasons set forth above, it is respectfully requested that the rejection of claims 1-5 and 7-13 under 35 USC 103(a) over the teachings of the cited references be withdrawn, as the combination of this myriad of patents fail to render claims 1-5 and 7-13 obvious.

Conclusion

In view of the foregoing arguments and amendments, Applicant believes that the application meets all applicable statutory and regulatory requirements. Accordingly, Applicant respectfully requests allowance of all claims remaining in the application.

If the Examiner has any questions regarding this amendment and/or believes that a telephone interview would assist in the advancement of this case to allowance, he/she is invited to contact the undersigned Agent for Applicant.

Respectfully submitted,

by Michele K. Yoder

Michele K. Yoder
Reg. No. 41,562
Agent for Applicant

JAMES RAY & ASSOCIATES
2640 PITCAIRN ROAD
MONROEVILLE, PA 15146-3309

TELEPHONE: (412) 380-0725
FACSIMILE: (412) 380-0748